Feature Selection for fault detection systems : application to the Tennessee Eastman Process.

Brigitte Chebel-Morello, Simon Malinowski, Hafida Senoussi

Applied Intelligence manuscript No.
(will be inserted by the editor)

Feature Selection for fault detection systems : application to the Tennessee Eastman Process

Brigitte Chebel-Morello · Simon Malinowski · Hafida Senoussi

the date of receipt and acceptance should be inserted later

Abstract In fault detection systems, massive amount of data gathered from the life-cycle of equipment is often used to learn models or classifiers that aims at diagnosing different kind of errors or failures. Among this huge quantity of information, some features (or sets of features) are more correlated with the kind of failures than others. The presence of irrelevant features might affect the performance of the classifier. To improve the performance of a detection system, feature selection is hence a key step. We propose in this paper an algorithm named STRASS, that aims at detecting relevant features for classification purposes. In certain cases, when there exists a strong correlation between some features and the associated class, classical feature selection algorithms fail at selecting the most relevant features. In order to cope with this problem, STRASS algorithm makes use of k-way correlation between features and the class to select relevant features. To assess the performance of STRASS, we apply it on simulated data collected from the Tennessee Eastman chemical plant simulator. The Tennessee Eastman process (TEP) has been used in many fault detection studies and three specific faults are not well discriminated with classical algorithms. The results obtained by STRASS are compared to those obtained with reference feature selection algorithms. We show that the features selected by STRASS always improve the performance of a classifier compared to the whole set of original features and that the obtained classification is better than with most of the other feature selection algorithms.

Keywords Feature Selection · Wrapper method · Fault detection · Contextual measure

1 Introduction

Fault detection has been extensively studied over the last few decades using various techniques. Tyan et al. [36] give a retrospective of the different methods such as parameter

B. Chebel-Morello and S. Malinowski
FEMTO-ST/ENSMM, Besanon, France
E-mail: brigitte.morello@femto-st.fr, simon.malinowski@femto-st.fr

H. Senoussi
Univ. of Sciences and Technology, Mohamed Boudiaf, Oran
E-mail: senoussih@yahoo.fr

estimation, state observation schemes, pattern recognition techniques and artificial intelligence methods. Fault detection techniques can be divided into three categories: model-based, knowledge-based [30, 8] and data-driven methods [37, 36, 32, 35, 39]. The approach proposed in this paper is data-driven and is based on machine learning tools.

A fault detection system, as shown in Fig. 1, can be identified from the knowledge data discovery process (KDD) where the output gives the state of the system (faulty or non-faulty for instance). The input data correspond to recorded sensor measurements that are considered as features.

The data-mining component relies heavily on classical techniques from the fields of machine learning, pattern recognition, and statistics to find relevant patterns from the data and transform them into useful task-oriented knowledge.

[Fig. 1 about here.]

Knowledge discovery and data mining have emerged as some of the most significant and fast expanding research areas and have found many successful real-world applications in a variety of disciplines like fault detection. For instance, Casimira at al. [7] develop a K-nearest neighbors algorithm to identify stator and rotors faults in induction motors. From the thirty-one features extracted by time-frequency analysis of the stators currents and voltages, six relevant features are selected by a sequential backward algorithm from the initial subset. Experimental results demonstrated the effectiveness of this method in condition monitoring. Sugumara at al. [32] focused particularly on fault conditions in roller bearings of a rotary machine. They used vibration signals from different functional mode (good bearing, bearing with inner race fault, bearing with outer race fault, and inner and outer race fault). First, a set of eleven features were extracted by time-frequency analysis. Among these, four best features were selected from a given set of samples using C4.5 decision algorithm. Secondly, Proximal Support Vector Machine (PSVM), was used to efficiently classify the faults. Yang et al [40] presents a survey of fault diagnosis using Support Vector Machines (SVM) combined with other methods. In a similar study on fault detection of roller bearings, Jack at al. [16] used Genetic Algorithms to select an optimal feature subset for SVM and artificial neural network based classifiers.

In chemical process industry large amounts of variables are monitored, which makes feature selection an important topic for that kind of applications. The Tennessee Eastman Process (TEP) benchmark has been the object of many studies in the literature [12, 29]. L. Wang and J. Yu have proposed in [38] a binary Particle Swarm Optimization with mutation (MBPSOM) combined with Support Vector Machine (SVM) to select the most pertinent for fault diagnosis. Chiang et al. [9] applied Fisher Discriminant Analysis (FDA), Discriminant Partial Least Squares (DPLS) and Principal Component Analysis (PCA). Nashalji et al. [27] used Genetic Algorithm and PCA to determine the main principle components and then used a neural networks-based classifier to detect faults during the operations of the industrial process. Verron et al. [37] proposed a fault diagnosis procedure based on discriminant analysis and mutual information. In order to obtain accurate classification performance, feature selection is performed with an algorithm based on the mutual information between variables. P. Cui et al. [10] applied Kernel principal component analysis (KPCA) for analysis of the TEP, and improved the fault detection of KPCA by a Fisher discriminant analysis scheme.
In the TEP benchmark, three kinds of fault (4,9 and 11) are difficult to distinguish because they are strongly correlated. To improve the efficiency of fault detection systems with strongly correlated data, we develop in this paper a detection system based on a feature selection algorithm devoted to detect interactions between the features and the class. The

presence of irrelevant and/or redundant features affects the speed and accuracy of learning or data mining algorithms. Therefore the selection of relevant information is very important for the development of a comprehensive and robust model and also for speeding up the training phase, hence reducing the cost as well as the data collection time for the classifier [2, 5, 11, 22].

This paper takes up the feature interaction challenge. We are specifically interested in detecting partial correlations among variables. Two criteria are proposed (see section 3) respectively based on the weak and the strong relevance: the discriminating capacity measure (DC) and the discriminating capacity gain measure (DCG). These criteria are designed to focus on detecting k-way interactions in the data (i.e., interactions between sets of k features and the class) and therefore on detecting features that have the exclusiveness to discriminate concepts (also calles unavoidable features). These criteria are established in a greedy type algorithm named STRASS (STrong Relevant Algorithm of Subset Selection). This algorithm has the ability to treat partially correlated data. In order to highlight the effectiveness of the proposed algorithm, and assess its capability to detect partial correlations, STRASS is first tested on artificial data sets which are well known for their feature interactions. Indeed, those data sets have challenged many feature selection algorithms. STRASS is then applied to the TEP benchmark data, by feeding different classifiers with the set of features selected by STRASS. Experimental results are compared with reference algorithms as CFS, FCBF, mRMR, INTERACT, LASSO, ReliefF and SVM-RFE. They highlight the ability of the proposed algorithm to select relevant features for classification purposes.

The rest of the paper is organized as follows: Section 2 overviews some of the feature selection methods used for fault detection. Section 3 introduces the selection criteria uponn which STRASS is built. A comprehensive description of STRASS is presented in Section 4. In Section 5, STRASS is evaluated on synthetic data sets and on the TEP benchmark and compared with well-known feature selection algorithms. Conclusions are drawn in Section 6.

2 Overview of feature selection methods

Different feature selection strategies have been proposed and used in the fault detection context. Liu [22, 23], Blum and Langley [5] compared different approaches dealing with data selection and emphasized four major points: the starting point in the feature space, the search strategy, the evaluation criterion and the stopping criterion.

- The starting point in the feature space could begin with no features with a forward search, all features with a backward search, or a random subset of features with a bidirectional search. Consequently, features can be successively added or removed by a certain procedure.
- The search strategy for feature subsets can be done by random heuristics or complete procedure.
- The evaluation criteria is an important component of any feature selection method, as it measures the goodness of a specific subset. An evaluation criteria can be categorized into three main groups based on their dependency on mining algorithm: filters, wrappers and embedded methods. Filters operate independently of any mining algorithm contrary to wrapper methods which use the performance of the mining algorithm. Embedded methods are built upon a data mining or a classification algorithm to perform the feature selection.

We propose to categorize feature selection algorithms depending of their evaluation criteria into three main groups based on how the interaction between features is treated [25]:

- The myopic measures which estimate the feature quality independently of the others features. Most of the existing measures belong to the first category that is why Kira and Rendel [17] and Kononenko [18] underlined the difficulties for classifiers to work with correlated data.
- Semi-contextual measures consider low order 2-way (one feature and the class) and 3-way (two features and the class) interactions.
- Contextual measures consider k-way (k features and the class, $k > 2$) interactions.

2.1 The myopic measures

Algorithms based on myopic criteria Relief [17], B&B [26], LVF [11], FOCUS [3] do not detect the correlations between the features and the class, unlike those using semi-contextual or contextual criteria. Indeed, most of the works in statistics make the features independence assumption. Relief, the most powerful individual feature selection algorithm, scores individual features rather than scoring feature subsets, those features with scores exceeding a user-specified threshold are selected for the final subset. The actual challenge in feature selection is to study feature interactions with relevant measures to select the optimal subset with maximum relevance and minimum redundancy.

2.2 Semi-contextual measures

CFS [15], mRMR [28], FCBF [40] and ReliefF [18, 19] use semi contextual measures. CFS calculates a feature subset merit, thus detecting the best feature combination. The algorithm is powerful as long as the interaction between features is not too large. mRMR feature selection algorithm selects features that should be both minimally redundant among themselves and maximally relevant to the target classes. The optimal subset maximizes the distance between the two profits. FCBF uses a correlation measure based on the information gain to detect the redundancy between features. They chose the Symmetrical Uncertainty (SU). The algorithm involves two steps: the first one calculates the SU value for each features, selects and orders relevant ones according to a predefined threshold, and the second one selects predominant features. However none of these algorithms treats high order interactions (k features and the class) like a contextual measure. Genetic Algorithm (GA) based feature selection algorithms [6, 21] have been successfully used as a variable subset selection and attempt to address the variable associations and various effects. Feature interaction is indirectly taken into account via the selection of a set of variables which is determined by a fitness function, generally based on the classifier accuracy. However, it should be noted that this method is a wrapper method unlike other methods. The LASSO method [34] is an embedded algorithm. It constructs a model and penalizes coefficients, shrinking many of them to zero. The principle of LASSO is linked with L1-norm regularization techniques which aim at penalizing complex models, upon which is also built the Elastic Net approach [42]. SVM-RFE [13] is also one of the most famous embedded method. Weights are assigned to features while building the model, and the ones with smaller weights are recursively eliminated.

2.3 Contextual measures

Two more recent algorithms treat and capture k-way interacting features : INTERACT [41] and STRASS [8, 25, 31]. To do so, INTERACT combines an information measure and a consistency measure. In the first part of the algorithm, the features are ranked in descending order based on their symmetrical uncertainty (SU) values. In the second part, features are evaluated according to their C-contribution which relies on the calculation of inconsistency rate. The features are evaluated one by one starting from the end of the ranked feature list. The strong points of these algorithms are their effectiveness to deal with diverse problems like, modal, continuous, noisy and correlated data.

3 Criteria of relevance and redundancy

In this section, we define two new criteria for feature selection, that are elaborated from the class discriminatory power, in a pair-wise data representation approach. The STRASS feature selection algorithm (cf Section 4) is built upon these criteria. These criteria are designed to take into account k-way interactions in the data and hence lead to a contextual measure.

3.1 Data representation

Let the input data Ω consist of n samples $\omega_1, \dots \omega_n$. Every sample in Ω is composed of r features. The set of features is denoted $x = \{x_1, \dots, x_r\}$. Every sample in Ω is labeled with a class $c \in \mathscr{C} = \{c_1, \dots, c_M\}$. In the following, the notation $x_k(\omega_i)$ represents the value of feature x_k in ω_i and $C(\omega_i)$ represents the class of sample ω_i. Let us associate to a feature x_k the Boolean function ϕ_{ij}^k, $1 \leq i, j \leq n$, $1 \leq k \leq r$:

$$\begin{array}{rl} \phi_{ij}^k : \Omega \times \Omega & \to \{0,1\} \\ (\omega_i, \omega_j) & \mapsto 1 \text{ if } x_k(\omega_i) = x_k(\omega_j) \\ & \quad\ 0 \text{ otherwise.} \end{array} \tag{1}$$

Let us also define the function $\phi_{ij}^{\mathscr{C}}$:

$$\begin{array}{rl} \phi_{ij}^{\mathscr{C}} : \Omega \times \Omega & \to \{0,1\} \\ (\omega_i, \omega_j) & \mapsto 1 \text{ if } C(\omega_i) = C(\omega_j) \\ & \quad\ 0 \text{ otherwise.} \end{array} \tag{2}$$

3.2 Weak Relevance measure

The weak relevance of a set of features is defined by the number of all pairs of samples who have at least one discriminating variable and different labels or different distributions of labels. According to that definition, let us define the discriminating capacity (DC) measure of a feature set $L = (x_1, \dots, x_m)$ with the following formula :

$$DC(L, \Omega) = \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{\prod_{k=1}^{m} \phi_{i,j}^k} \bullet \overline{\phi_{i,j}^{\mathscr{C}}} \tag{3}$$

3.3 Strong Relevance measure

To measure the exclusiveness of a feature to describe a concept, the equivalent of a "relevance gain" is defined as the measure related to a feature compared to a subset of features.

The strong relevance (SR) of a feature x_k on pairs of instances is defined as the relevance of a feature x_k compared to a relevant preselected features subset $L = (x_1, \ldots x_m)$. This measure is given by:

$$SR(x_k, L, \omega_i, \omega_j) = \overline{\phi_{i,j}^{\mathscr{C}}} \bullet \overline{\phi_{i,j}^{k}} \bullet \prod_{l=1}^{m} \phi_{i,j}^{l} \tag{4}$$

The aggregation of the Strong Relevance (SR) expression on the whole pairs obtained by the sample Ω of n patterns will define the Discriminating Capacity Gain (DCG) as:

$$DCG(x_k, L, \Omega) = \sum_{i=1}^{n} \sum_{j=1}^{n} SR(x_k, L, \omega_i, \omega_j) \tag{5}$$

The DCG of a feature x_k for a set of n objects compared to a set L of features is equal to the number of object couples discriminated by only x_k and no other features.

3.4 Redundancy of a feature

A feature x_k is said to be redundant in a feature subset L if the discriminating capacity measure of the set $L \backslash \{x_k\}$ is the same as the one of L. In other words, x_k is said to be redundant if

$$DC(L, \Omega) = DC(L \backslash \{x_k\}, \Omega) \tag{6}$$

3.5 Interest of the two criteria

The two suggested contextual criteria upon which STRASS is built are complementary. They can detect not only strongly relevant features but also weakly relevant ones. The first one calculates the discriminating capacity of a set of variables, and aims at extracting a subset of variables with the same DC as the entire set. The second one evaluates the discriminating capacity gain of one feature relatively to a set of features. This contextual criterion aims at detecting either the relevance or the redundancy of a feature compared to a subset of features. Features with the largest gains are integrated into the selected set, while the redundant ones (with a null gain) are discarded. These criteria have the particularity to be calculated on a restricted set of object pairs and variables. It can hence detect partial correlations between features, and study the k-way interactions between them and the class. The criteria which we have developed under pair data set allow us to explore three aspects of the correlation:

1. the feature correlation on a pair-wise data set, i.e. the feature capacity to discriminate a part of the studied population.
2. the partial feature correlation relatively to a set of features. Redundant features (that play the same role of another feature) are searched to be excluded. On the other hand, a feature can be considered weakly relevant to the class when evaluated alone, but becomes very relevant when combined with other features. This correlation is also searched with these criteria.
3. the feature capacity to be the only one to discriminate a population subset. Such features are called strongly relevant (or predominant).

4 STRASS (STrong Relevant Algorithm of Subset Selection

4.1 Description of the algorithm

STRASS is based on the contextual criteria established in Michaut thesis [25, 8], and described in the previous section. The first criterion measure is a discriminating capacity (DC) of a set of variables and the second criterion is DCG (Discriminating Capacity Gain) measure. These criteria when associated with a greedy algorithm allow:

- To capture k-way interacting features
- To detect the partially redundant features. Partial correlation exists when only a feature combination can discriminate the class
- To rank the variables selected with the complementary criteria (DC: discriminating capacity).

The STRASS algorithm proposed here belongs to the greedy type category of algorithms. The research is a sequential bidirectional generation, i.e. a core of features is composed from an empty set S_f which is built gradually until a subset having the same degree of relevance as the starting subset noted is obtained. The feature subset is progressively computed and re-evaluated at every feature addition. The algorithm breaks up into three steps depending on its initialization:

Step 1: The features are ranked in descending order based on their discriminating capacity gain and a subset of strongly relevant features or predominant features is selected.

Step 2: The remaining features are evaluated one by one starting from the top of the remaining features list and the weakly relevant features which have the largest discriminating capacity are combined with the previously selected features S_f if the resulting overall discriminating power is increased. In fact a feature may have a little correlation with the class, but when it is combined with some others features, the resulting subset can be strongly correlated with the class.

Step 3: Suppression of redundant features. At this stage, backward elimination is employed to detect the features that become redundant compared to the preselected features subset S_f when adding a new feature in the second stage of the algorithm. For a predefined threshold $\rho, 0 < \rho < 1$ features having a discriminating capacity $DC < \rho \times DCtot$ are removed. Therefore, we obtain a subset of low cardinal and with no redundancy in the selected set of features.

The complete algorithm is given in Figure 2.

4.2 Strengths and weaknesses of STRASS

Amongst feature selection algorithms, STRASS is based on contextual criteria. These criteria are established in a greedy type algorithm and select from a learning set a minimal set of relevant features. The learning set must be consistent, not having missing data and consists of symbolic data and/or numeric. Noisy data have a negative impact on the associated performance. To our knowledge, the criteria used by STRASS are the first to take into account the different aspects of the correlation detailed above. STRASS identifies the k-way interaction between features and the partial correlations and partial redundancy in a set of pairwise

data. Another interesting remark is the fact that STRASS is a filter algorithm which is less computationally intensive than wrapper techniques, and embedded methods.

STRASS is computationally efficient on databases with reasonable sizes (in the order of thousands entries and hundreds of features). However for large databases, the pairwise data representation is inherently a combinatorial problem, and is not adapted. In order to reduce the complexity of STRASS, we plan to simplify the criteria and express them under a contingency form. The transformation of the pairwise criteria to contingency criteria with the help of Marchotorchino [24] might be of interest and will be the object of a future work.

[Fig. 2 about here.]

5 Experiments and results

5.1 Implementation

STRASS algorithm was implemented in MATLAB 7.5 environment. For the filtering algorithms and classifiers, existing tools in WEKA machine learning platform [14] have been used. The experiments were run using WEKA with its default values. For evaluation purposes we will compare STRASS with other algorithms that consider feature interaction and correlation using contextual and semi-contextual measures, such as CFS, mrMR, CFS-FCBF, INTERACT, ReliefF, SVM-RFE, LASSO, Elastic Nets. The mRMR feature selection algorithm used in this work has been adapted from [28] and has been downloaded online[1]. Same conditions are used in Matlab and Weka.

5.2 Evaluation of the algorithm

STRASS performance was assessed in two complementary ways: (i) Direct evaluation through artificial data sets: Led, Monk, Bool, Parity, Corral and Agrawal's functions (Appendix B). These data sets highlight the behavior of our algorithm when the descriptive characteristics interact. Langley and Sage [20] stressed the importance of artificial fields. (ii) Indirect evaluation on TEP benchmark allows us to study the classification performance with and without the filtering phase, the classifiers accuracy and the number of features removed by the filtering algorithm. The results obtained with STRASS are compared with filter methods, as mRMR, CFS (with best first search), FCBF (with threshold SU set to 0), INTERACT, ReliefF algorithms, and with embedded methods, as the LASSO principle [34], the Elastic Net principle [42] the SVM-RFE algorithm [13]. To compare STRASS with mRMR and ReliefF, we take the same number of features for mRMR as the one selected by STRASS.

5.3 Synthetic data with known feature interaction

[Table 1 about here.]

We have applied STRASS on some synthetic data sets widely used for feature selection. Results are given in Table 1. Only STRASS is able to determine relevant features in all data

[1] http://penglab.janelia.org/software/

sets. The INTERACT algorithm selects relevant features nine times out of thirteen, which is more than for the other algorithms. It can be seen from Table 1 that for LED Display Domain (Led, Led 24) all the algorithms fail to detect the relevance and the sufficiency of the five segments except STRASS. For the Parity data set, STRASS and INTERACT establish that the features x_1, x_2, x_3 are relevant whereas the others are useless. For the three MONKs data sets, STRASS was able to find true relevant features using a loss of 1% of the $DCtot$ for MONK-3. STRASS, INTERACT and mRMR detected the redundancy of feature x_6 in Corral data set. This feature being correlated to 75% with the feature class is considered to be relevant alone by CFS and FCBF because the algorithms cannot evaluate the k-way interactions. In the case of Agrawal's functions (F1 to F4), STRASS gives the relevant features for the four functions. Thus, STRASS is the most powerful algorithm on these data sets. This is explained by the presence of large interactions in the data. Indeed CFS, mRMR and FCBF detect the pair-wise feature-feature correlation (inter-correlation) but they cannot identify neither interactions between subsets of features and the class nor unavoidable features.

5.4 Application on the TEP Benchmark

[Table 2 about here.]

TEP benchmark [12] is described in Appendix B. Instances in this data set are composed of 52 variables and labeled with a fault type between 1 and 15. Three types of faults denoted as fault 4, fault 9 and fault 11 are considered here, because they are the most difficult to classify. The problem of their identification is due to a great interaction between the features for these faults, not to the classifier used. To solve this problem and improve the classifier performance, we need define a feature filtering taking account feature interactions. Table 2 presents the features that are selected by the different algorithms compared here. For STRASS, the most discriminating features are $\{51, 41, 38, 40, 37, 50, 19, 18, 9, 20\}$. STRASS uses DCG (discriminatory capacity gain) to rank each feature are sorted according to its contribution relatively with other features. Fig. 3 shows the DCG for each predominant feature.

[Fig. 3 about here.]

[Table 3 about here.]

For the classification task, five different classifiers have been used : a decision tree (C4.5), 1-nearest-neighbor (IB1), Naive Bayes (NB), a multilayer perceptron (MLP) and Support Vector Machines (SVM). We have applied these classifiers to the features selected by the feature selection algorithms described above. Results are obtained with 10-fold cross validation. We have compared the results with semi-contextual and contextual methods : mRMR, CFS, FCBF, INTERACT, ReliefF, LASSO, Elastic net and the SVM-RFE feature selection algorithms. To analyze the results obtained in this study, we have employed two performance measures: accuracy and Cohen's Kappa. The accuracy is defined as the number of successful hits relative to the total number of classifications. The Kappa is a statistical measure of inter-rater agreement. Cohen's Kappa can be adapted to classification tasks and it is also used in some well-known software packages, such as WEKA. Tables 3 and 4 show the performance in terms of accuracy and Kappa respectively. The best results for each classifier are highlighted in bold. The symbols $+$ and $-$ respectively indicates an improvement

or a degradation in terms of performance compared with the ones obtained for the whole set of features.
Let us examine the effect of feature selection on classification performance. Classification accuracy and kappa score is calculated before and after filtering. These results are reported in Tables 3 and 4. An interesting point is that the features selected by STRASS always improve the performance of a classifier compared to the complete set of features. This is due to the fact that the feature subset selected by STRASS has the same discriminating capacity as the full set of features and hence the classification performance are equal or better than the one of the full set. The performance obtained with STRASS is always one of the two best feature selection algorithms apart from when using with the Multi-Layer perceptron. Its average position is 2.6 which is the best among the other algorithms. The best performance in terms of accuracy is obtained with IB1 and ReliefF algorithms (98.91%), but STRASS is just behind that performance (98.56%). Moreover, the feature selection made by the ReliefF algorithm does not always improve the performance compared to the whole features on the contrary to STRASS.

In terms of average classification performance, the best algorithms are STRASS, mRMR and INTERACT. These results hence highlight the benefit of detecting k-way correlations (particularly for $k < 2$, as STRASS and INTERACT) compared to only detecting feature-feature intercorrelation, as most of the semi-contextual and embedded consider. The characteristic of detecting k-way correlations lead to improve classification performance each time on these experiments.

[Table 4 about here.]

[Fig. 4 about here.]

Fig 4 gives the accuracy results obtained with STRASS with ordered selected features in decision tree (C4.5) and nearest neighbors classifier (IB1). For IB1 the best results are obtained with all the selected features whereas for C4.5 the seven first selected features give the best result.

The confusion matrices obtained with STRASS + IB1 and ReliefF + IB1 for their associate best feature set are given respectively in Table 5 and 6. In all these cases, the fault 11 is less discriminated because this fault overlaps with the two others.

To conclude this experiment section, we have seen that STRASS gives better or equivalent performance than most of the compared feature selection algorithms when combined with different classifier. An interesting point is that the features selected by STRASS always improve the performance of a classifier compared to the complete set of features, which is not always the case for the other feature selection algorithms. These results hence highlight the benefit of the k-way feature selection process of STRASS.

[Table 5 about here.]

[Table 6 about here.]

6 Conclusion

This paper describes STRASS, a contextual-based feature selection algorithm for classication purposes able to detect the interaction between features and the class and select a minimum relevant feature subset. The efficiency and effectiveness of STRASS to handle large interactions are demonstrated through a comparative study with other representative

feature selection algorithms on synthetic data known for their correlation. The proposed feature selection algorithm was then applied to a well-known fault detection benchmark: the Tennessee Eastman Process (TEP). STRASS has demonstrated its ability to reduce the dimensionality of data sets while maintaining or improving the performances of learning algorithms. In fact for TEP process, the features selected by STRASS decreased the data correlation and the overall misclassification for the testing set using 1-nearest neighbor decreased further to 1.4%. STRASS was also compared to other reference feature selection algorithms. The results of STRASS outperformed or lead to equivalent performance to those obtained with those reference algorithms. In addition, the features selected by STRASS always improve the performance of a classifier compared to the whole set of features.

A The Tennessee Eastman Process

[Table 7 about here.]

The Tennessee Eastman Process (TEP) is a chemical process, created by the Eastman Chemical Company to provide a realistic industrial process in order to evaluate process control and monitoring methods [12]. This process was simulated on Matlab by Ricker [29]. The simulator was used to generate overlapping data sets to evaluate the classification performance. Figure 5 shows a flow sheet of TEP. There are four unit operations: an exothermic two phase reactor, a flash separator, a re-boiler striper, and a recycle compressor. The TEP process produces two products (G and H) and one (undesired) by-product F from four reactants (A, C, D and E). This process has 12 input variables and 41 output variables. Only 52 variables are taken into account in this problem because one of the input variables (the reactor agitator speed) is constant. The system has fifteen types of identified faults. In this paper, we considered only three types of fault : fault 4, 9 and 11. These faults are described in Table 7.

[Fig. 5 about here.]

B Synthetic data

We describe in this appendix the synthetic data used in this paper for simulation purposes.
The **LED display domain** data set is available on the UCI data set repository [4].
The **MONK's** problems [33] are composed of three target concepts :
MONK-1 : $(x_1 = x_2) \vee (x_3 = 1)$
MONK-2 : exactly two of :

$$\{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1\}$$

MONK-3 : $(x_5 = 3 \wedge x_4 = 1) \vee (x_5 \neq 4 \wedge x_2 \neq 3)$
The **BOOL** data set is composed of a function of 6 Boolean features giving a Boolean class, for instance : $y_{class} = (x_1 \oplus x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$. Six other randomly generated Boolean features are added to these features.
The **Parity** data set is composed of a of a function of 3 Boolean features $y_{class} = x_1 \oplus x_2 \oplus x_3$. Seven randomly generated Boolean features are added. This data set is particularly interesting because no relevant features taken separately can be distinguished from irrelevant ones.
The **Parity2** data set is the same as the **Parity** data set to which 2 redundant features are added : $x_{11} = x_1$ and $x_{12} = x_2$. This data set allows testing the algorithms ability to work with redundant features.
The **Coral** data set is composed of six binary features x_1 to x_6 among which x_5 is irrelevant and x_6 is correlated to 75% with the feature class $y_{class} = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$.
Agrawal's functions are a series of classification functions of increasing complexity that uses 9 features to classify people into different groups. More details can be found in [1].

References

1. Agrawal R, Ghosh S, Imielinski T, Iyer B, Swami A (1992) An interval classifier for database mining applications. In: Proceedings of the 18th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, VLDB '92, pp 560–573
2. Almuallim H, Dietterich TG (1991) Learning with many irrelevant features. In: In Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI Press, pp 547–552
3. Almuallim H, Dietterich TG (1994) Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence 69:279–305
4. Bache K, Lichman M (2013) UCI machine learning repository
5. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence 97:245–271
6. Casillas J, Cordn O, Jesus MJD, Herrera F, Casillas J, Herrera F (2000) Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems
7. Casimir R, Boutleux E, Clerc G, Yahoui A (2006) The use of features selection and nearest neighbors rule for faults diagnostic in induction motors. Engineering Applications of Artificial Intelligence 19(2):169 – 177
8. Chebel Morello B, Michaut D, Baptiste P (2001) A knowledge discovery process for a flexible manufacturing system. In: Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on, pp 651–658 vol.1
9. Chiang LH, Kotanchek ME, Kordon AK (2004) Fault diagnosis based on fisher discriminant analysis and support vector machines. Computers & Chemical Engineering 28(8):1389 – 1401
10. Cui P, Li J, Wang G (2008) Improved kernel principal component analysis for fault detection. Expert Systems with Applications 34(2):1210 – 1219
11. Dash M, Liu H, Motoda H (2000) Consistency based feature selection. In: Terano T, Liu H, Chen A (eds) Knowledge Discovery and Data Mining. Current Issues and New Applications, Lecture Notes in Computer Science, vol 1805, Springer Berlin Heidelberg, pp 98–109
12. Downs J, Vogel E (1993) A plant-wide industrial process control problem. Computers & Chemical Engineering 17(3):245 – 255
13. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3):389–422
14. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: An update. SIGKDD Explor Newsl 11(1):10–18
15. Hall MA (2000) Correlation-based feature selection for discrete and numeric class machine learning. Morgan Kaufmann, pp 359–366
16. Jack L, Nandi A (2000) Genetic algorithms for feature selection in machine condition monitoring with vibration signals. Vision, Image and Signal Processing, IEE Proceedings - 147(3):205–212
17. Kira K, Rendell LA (1992) The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI Press, AAAI'92, pp 129–134
18. Kononenko I (1994) Estimating attributes: Analysis and extensions of relief. Springer Verlag, pp 171–182
19. Kononenko I, Simec E, Robnik-Sikonja M (1997) Overcoming the myopia of inductive learning algorithms with relieff. Applied Intelligence 7:39–55
20. Langley P, Sage S (1997) Computational learning theory and natural learning systems: Volume iv. MIT Press, Cambridge, MA, USA, chap Scaling to Domains with Irrelevant Features, pp 51–63
21. Lanzi PL (1997) Fast feature selection with genetic algorithms: a filter approach. In: Evolutionary Computation, 1997., IEEE International Conference on, pp 537–540
22. Liu H, Motoda H (1998) Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Norwell, MA, USA
23. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. Knowledge and Data Engineering, IEEE Transactions on 17(4):491–502
24. Marcotorchino F (1984) Utilisation des comparaisons par paires en statistique des contingences. Centre scientifique IBM Paris Etudes F-069, F-071, F-081
25. Michaut D (1999) Filterign and variable selection in learning processes. PhD, Univ of Franche Comt
26. Narendra PM, Fukunaga K (1977) A branch and bound algorithm for feature subset selection. IEEE Trans Comput 26(9):917–922
27. Noruzi Nashalji M, Aliyari Shoorehdeli M, Teshnehlab M (2010) Fault detection of the tennessee eastman process using improved pca and neural classifier. In: Gao XZ, Gaspar-Cunha A, Kppen M, Schaefer G, Wang J (eds) Soft Computing in Industrial Applications, Advances in Intelligent and Soft Computing,

vol 75, Springer Berlin Heidelberg, pp 41–50
28. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27:1226–1238
29. Ricker NL (1996) Decentralized control of the tennessee eastman challenge process. Journal of Process Control 6(4):205 – 221
30. Riverol C, Carosi C (2008) Integration of fault diagnosis based on case-based reasoning principles in brewing. Sensing and Instrumentation for Food Quality and Safety 2(1):15–20
31. Senoussi H, Chebel-Morello B (2008) A new contextual based feature selection. In: Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pp 1265–1272
32. Sugumaran V, Muralidharan V, Ramachandran K (2007) Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. Mechanical Systems and Signal Processing 21(2):930 – 942
33. Thrun S, Bala J, Bloedorn E, Bratko I, Cestnik B, Cheng J, Jong KD, Dzeroski S, Hamann R, Kaufman K, Keller S, Kononenko I, Kreuziger J, Michalski R, Mitchell T, Pachowicz P, Roger B, Vafaie H, de Velde WV, Wenzel W, Wnek J, Zhang J (1991) The MONK's problems: A performance comparison of different learning algorithms. Tech. Rep. CMU-CS-91-197, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA
34. Tibshirani R (1994) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58:267–288
35. Torkkola K, Venkatesan S, Liu H (2004) Sensor selection for maneuver classification. In: In proceedings of the 7th IEEE International ITSC Conference
36. Tyan CY, Wang PP, Bahler DR (1996) An application on intelligent control using neural network and fuzzy logic. Neurocomputing 12(4):345 – 363
37. Verron S, Tiplica T, Kobi A (2008) Fault detection and identification with a new feature selection based on mutual information. Journal of Process Control 18(5):479 – 490
38. Wang L, Yu J (2005) Fault feature selection based on modified binary pso with mutation and its application in chemical process fault diagnosis. In: Wang L, Chen K, Ong Y (eds) Advances in Natural Computation, Lecture Notes in Computer Science, vol 3612, Springer Berlin Heidelberg, pp 832–840
39. Widodo A, Yang BS (2007) Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors. Expert Systems with Applications 33(1):241 – 250
40. Yang BS, Widodo A (2008) Support Vector Machine for Machine Fault Diagnosis and Prognosis. Journal of System Design and Dynamics 2:12–23
41. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224
42. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B 67:301–320

List of Figures

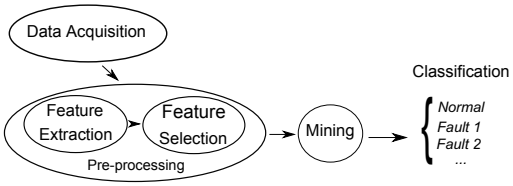

Fig. 1 Knowledge data discovery process

Input:
E the whole set of pairs
$S_o = \{x_1, \ldots, x_r\}$ the set of features
$DCtot = DC(S_o, E)$ DC of S_o
ρ threshold for loss of DC
Output:
S_f selected features
Initialize S_f to $\emptyset$
1. Selection of strongly predominant features
For each $x_k \in S_o$ **do**
  if $DCG(x_k, S_o - \{x_k\}, E) > 0$ **then**
    $S_f = S_f + \{x_k\}$
    $S_o = S_o \setminus \{x_k\}$
  end
end
Update E, as $E = E \setminus \{\text{discriminated pairs}\}$

2. Selection of the remaining weak relevant features
While $DC(S_f, E) < \rho \times DCtot$ **do**
  $DCmax = 0$
  For each $x_k \in S_o$ **do**
    if $DC(x_k, E) > DCmax$ **then**
      $DCmax = DC(x_k, E)$
      $x_{max} = x_k$
    end
  end
  $S_f = S_f + \{x_{max}\}$
  $S_o = S_o \setminus \{x_{max}\}$
end
Update E, as $E = E \setminus \{\text{discriminated pairs}\}$

3. Elimination of redundant features
For each $x_k \in S_o$ **do**
  if $DC(S_f \setminus \{x_k\}, E) = DC(S_f, E)$ **then**
    $S_f = S_f \setminus x_k$
  end
end

Fig. 2 Description of the STRASS algorithm

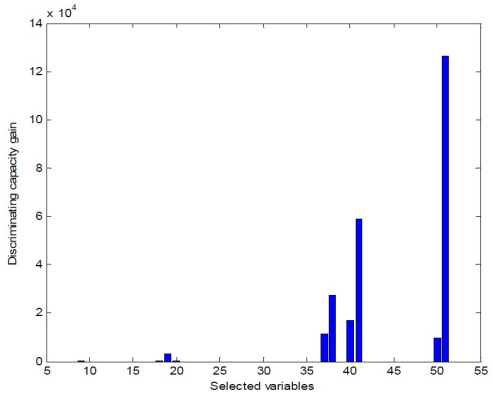

Fig. 3 Accuracy performance for ordered selected features with STRASS and two classifiers : IB1 and C4.5

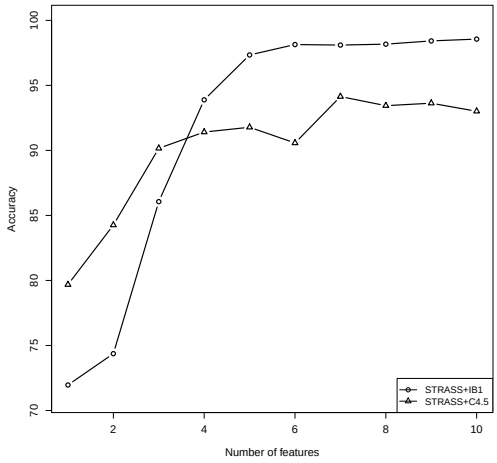

Fig. 4 Accuracy performance for ordered selected features with STRASS and two classifiers : IB1 and C4.5

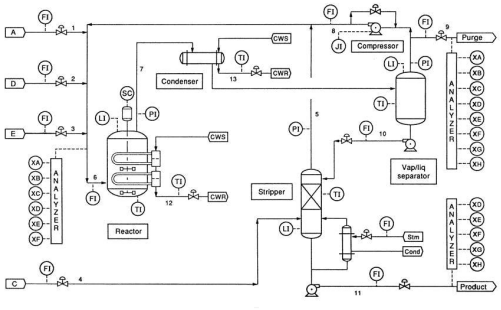

Fig. 5 Process flow sheet of TEP

List of Tables

Table 1 Features selected by different algorithms on synthetic data sets

data sets	Relevant features	STRASS	mRMR	CFS	FCBF	INTERACT	ReliefF
LED7	$x_{1:5}$	$x_{1:5}$	$x_{2:4}, x_{6:7}$	$x_{1:7}$	$x_{1:7}$	$x_{1:7}$	$x_{1:7}$
LED 24	$x_{1:5}$	$x_{1:5}$	$x_{2:5}, x_7$	$x_{1:8}, x_{14}$ x_{16}, x_{19}, x_{21}	$x_{2:7}, x_{11}$ x_{12}, x_{14}, $x_{15}, x_{19:21}$	$x_{1:9}, x_{11}$ $x_{14:20}$, $x_{22:23}$	$x_{1:7}$
MONK1	$x_{1:2}, x_5$	$x_{1:2}, x_5$	$x_1, x_{4:5}$	$x_1, x_{3:5}$	$x_1, x_{3:5}$	$x_{1:2}, x_5$	$x_{1:5}$
MONK2	$x_{1:6}$	$x_{1:6}$	$x_{1:6}$	$x_{4:6}$	$x_{4:6}$	$x_{1:6}$	$x_{1:6}$
MONK3	$x_2, x_{4:5}$	$x_2, x_{4:5}$	$x_2, x_{5:6}$	$x_2, x_{5:6}$	$x_2, x_{5:6}$	$x_{1:2}, x_{4:5}$	$x_{1:6}$
Parity	$x_{1:3}$	$x_{1:3}$	x_2, x_6, x_{10}	x_5, x_8, x_{10}	x_{10}	$x_{1:3}$	$x_{1:3}$
Parity 2	$x_{1:3}$	$x_{1:3}$	x_2, x_6, x_{10}	x_5, x_8, x_{10}	x_{10}	$x_{1:3}$	$x_{1:3}, x_{11:12}$
Corral	$x_{1:4}$	$x_{1:4}$	$x_{1:4}$	$x_{1:4}, x_6$	$x_{1:4}, x_6$	$x_{1:4}$	$x_{1:4}, x_6$
Bool	$x_{1:6}$	$x_{1:6}$	$x_{3:6}, x_7$, x_{10}	$x_{1:6}$	$x_{3:6}, x_{10}$ x_{12}	$x_{1:6}$	$x_{1:6}, x_7$, x_{12}
F1	x_3	x_3	x_3	x_3	x_3	x_3	x_3
F2	x_1, x_3	x_1, x_3	x_1, x_8	x_1	x_1	x_1	$x_{1:3}$
F3	$x_1, x_{3:4}$	$x_1, x_{3:4}$	x_2, x_4, x_8	$x_{2:4}$	$x_{2:4}$	$x_{2:4}$	$x_{1:9}$
F4	$x_{1:2}, x_9$	$x_{1:2}, x_9$	$x_{1:2}, x_8$	x_9	x_1, x_9	$x_{1:2}, x_8$	$x_{1:2}, x_9$

Table 2 Features selected by different algorithms on the TEP benchmark

STRASS	51,41,38,40,37,9, 50,18,19,20
mRMR	9,41,18,37,39,51,21,40,20,19
CFS	9,18,21,37,39,51
FCBF	9,18,21,37,39,51
Interact	51,9,41,38,37,50,40,19
ReliefF	51,37,40,39,41, 38,50,19,18,20
Lasso	51,9,28,32,3,39,35,43,40,34,11,29
SVM-RFE	28,35,34,9,51,19,18
Elastic Net	51,9,28,40,32,29,25,35

Table 3 Accuracy performance of different classifiers associated with feature selection algorithms

Classifier / FS Algo.	Decision Tree	IB-1	Naive Bayes	Multi-Layer Perceptron	SVM
None	90.34	84.21	86.71	85.95	44.08
STRASS	93.01+	98.56+	**87.06+**	86.23+	**83.02+**
mRMR	92.5+	97.18+	86.9+	**87.29+**	82.41+
CFS - FCBF	91.02+	88.19+	86.46−	86.64+	81.23+
INTERACT	**93.91+**	98.31+	86.97+	86.34+	80.39+
SVM-RFE	85.72−	90.6+	86.2+	87.04+	81.11+
LASSO	87.54−	85.67+	86.32−	86.34+	75.97+
ReliefF	90.42+	**98.91+**	82.61−	80−	82.98+
Elastic Net	87.1−	90.6+	86.27−	86.38+	80.37+

Table 4 Kappa score of different classifiers associated with feature selection algorithms

FS Algo. \ Classifier	Decision Tree	IB-1	Naive Bayes	Multi-Layer Perceptron	SVM
None	0.855	0.763	0.8	0.789	0.674
STRASS	0.895+	0.978+	**0.806+**	0.793+	**0.746+**
mRMR	0.886+	0.958+	0.803+	**0.809+**	0.736+
CFS - FCBF	0.865+	0.823+	0.797−	0.799+	0.718+
INTERACT	**0.909+**	0.975+	0.805+	0.795+	0.701+
SVM-RFE	0.7958−	0.859+	0.793−	0.805+	0.713+
LASSO	0.8132−	0.7851+	0.7948−	0.7951+	0.6396+
ReliefF	0.856+	**0.984+**	0.739−	0.7−	0.745+
Elastic Net	0.8066−	0.859+	0.7941−	0.7958+	0.7056+

Table 5 Confusion matrix obtained with STRASS+IB1

True Class \ Estimated class	Fault 4	Fault 9	Fault 11
Fault 4	1430	6	4
Fault 9	1	1435	4
Fault 11	27	20	1393

Table 6 Confusion matrix obtained with ReliefF+IB1

Estimated class / True Class	Fault 4	Fault 9	Fault 11
Fault 4	1432	1	7
Fault 9	1	1435	4
Fault 11	20	14	1406

Table 7 Description of the faults used in this paper

Fault number	Description
4	Step change in the reactor cooling water inlet temperature
9	Random variation in D feed temperature
11	Random variation in the reactor cooling water inlet temperature